Bioinformatics, 36(7), 2020, 2105–2112
doi: 10.1093/bioinformatics/btz863
Advance Access Publication Date: 18 November 2019
Original Paper

Structural bioinformatics

DeepMSA: constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant-homology proteins

Chengxin Zhang [1,†], Wei Zheng [1,†], S. M. Mortuza[1], Yang Li[1,2] and Yang Zhang[1,3,*]

[1]Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI 48109, USA, [2]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China and [3]Department of Biological Chemistry, University of Michigan, Ann Arbor, MI 48109, USA

*To whom correspondence should be addressed.

[†]The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

Associate Editor: Alfonso Valencia

Received on July 19, 2019; revised on October 17, 2019; editorial decision on November 14, 2019; accepted on November 15, 2019

Abstract

Motivation: The success of genome sequencing techniques has resulted in rapid explosion of protein sequences. Collections of multiple homologous sequences can provide critical information to the modeling of structure and function of unknown proteins. There are however no standard and efficient pipeline available for sensitive multiple sequence alignment (MSA) collection. This is particularly challenging when large whole-genome and metagenome databases are involved.

Results: We developed DeepMSA, a new open-source method for sensitive MSA construction, which has homologous sequences and alignments created from multi-sources of whole-genome and metagenome databases through complementary hidden Markov model algorithms. The practical usefulness of the pipeline was examined in three large-scale benchmark experiments based on 614 non-redundant proteins. First, DeepMSA was utilized to generate MSAs for residue-level contact prediction by six coevolution and deep learning-based programs, which resulted in an accuracy increase in long-range contacts by up to 24.4% compared to the default programs. Next, multiple threading programs are performed for homologous structure identification, where the average TM-score of the template alignments has over 7.5% increases with the use of the new DeepMSA profiles. Finally, DeepMSA was used for secondary structure prediction and resulted in statistically significant improvements in the Q3 accuracy. It is noted that all these improvements were achieved without re-training the parameters and neural-network models, demonstrating the robustness and general usefulness of the DeepMSA in protein structural bioinformatics applications, especially for targets without homologous templates in the PDB library.

Availability and implementation: https://zhanglab.ccmb.med.umich.edu/DeepMSA/.

Contact: zhng@umich.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Multiple sequence alignment (MSA), also called 'sequence profile', is designed to collect and align multiple homologous sequences of a query protein of interest. Since it contains rich information about the evolutionarily conserved positions and motifs, which cannot be derived from the query sequence alone, it has found fundamental usefulness in various bioinformatics studies. In protein structure prediction, e.g. the MSA is the primary component to derive local secondary structure (SS) features (Jones, 1999; Wu and Zhang, 2008), residue–residue contacts (Adhikari *et al.*, 2018; Hanson *et al.*, 2018; He *et al.*, 2017; Wang *et al.*, 2017) and homologous structural templates (Soding, 2005; Wu and Zhang, 2008; Zheng *et al.*, 2019); these are of critical importance for the full-length 3D structure constructions (Ovchinnikov *et al.*, 2018; Zhang *et al.*, 2018). In protein function annotations, the use of MSAs also has major impacts on the accuracy of Gene Ontology (Cozzetto *et al.*, 2016; Zhang *et al.*, 2017) and ligand-binding site (Gil and Fiser, 2019; Yang *et al.*, 2013) predictions.

Due to the critical importance of MSA, much attention has been paid to the development of various MSA and sequence profile construction methods. While PSI-BLAST is one of the most widely used

approaches to query-specific sequence profile generation (Altschul *et al.*, 1997), HHblits (Remmert *et al.*, 2012) from the HH-suite (Steinegger *et al.*, 2019) recently becomes popular for profile hidden Markov model (HMM) construction. Meanwhile, Jackhmmer and HMMsearch tools from the HMMER suite (Eddy, 1998) are common alternatives for the applications. Both lines of programs have been heavily used, especially for the contact predictions that are recently found critical for template-free (or *ab initio*) protein structure prediction (Ovchinnikov *et al.*, 2017; Schaarschmidt *et al.*, 2018; Wu *et al.*, 2011). Most recently, a hybrid MSA generation approach combining HHblits and Jackhmmer searches is shown to improve contact prediction by MetaPSICOV2 (Buchan and Jones, 2018). There was also evidence showing that MSAs collected from metagenome protein sequences can increase the coverage of sequence homologies and be useful for contact-assisted *de novo* structure prediction (Ovchinnikov *et al.*, 2017; Wang *et al.*, 2019).

Despite the importance of MSA construction, few standalone pipelines exist which can efficiently generate sensitive MSAs from a query input sequence, especially when multiple large sequence databases are involved. To address this urgent need, we developed and release DeepMSA, a new open-source program that constructs deep (in the sense of more sequences with a high diversity) and sensitive MSAs by merging sequences from three whole-genome and metagenome databases through a hybrid homology-detection approach. In this approach, HHblits from HH-suite 2.0.16 (Steinegger *et al.*, 2019) and Jackhmmer/HMMsearch, which were modified from HMMER 3.1b2 (Eddy, 1998) package to make the output format more compact in order to reduce file input/output, are used to perform homologous sequence search, and the alignments are further refined by a custom HHblits database reconstruction step. Large-scale benchmark experiments have showed that, compared to the widely used HHblits, PSI-BLAST and Jackhmmer programs, DeepMSA can consistently improve the accuracy of contact and SS predictions, and threading programs, which is particularly important for distant-homology proteins.

2 Materials and methods

2.1 Counting the number of effective sequences in MSAs

A common approach to quantify the homologous sequence coverage and/or alignment depth of an MSA is by counting the normalized number of effective sequence (Nf):

$$\mathrm{Nf} = \frac{1}{\sqrt{L}} \sum_{n=1}^{N} \frac{1}{1 + \sum_{m=1, m\neq n}^{N} I[S_{m,n} \geq 0.8]} \quad (1)$$

where L is the length of the query protein, N is the number of sequences in the MSA, $S_{m,n}$ is the sequence identity between the mth and nth sequences and $I_{[\ ]}$ is an Iverson bracket, i.e. $I[S_{m,n} \geq 0.8]$ equals to 1 if $S_{m,n} \geq 0.8$, and to zero otherwise. While current literature lacks consensus in terms of the ideal Nf for contact prediction, we optimize the Nf cutoff as 128 to attain accurate contact prediction, as discussed later. An example to illustrate the mathematical meaning of Nf is shown at Supplementary Figure S1.

2.2 DeepMSA pipeline for MSA construction

The MSA construction process in DeepMSA can be divided into three stages, which correspond to the searching of three sequence databases [Uniclust30 (Mirdita *et al.*, 2017), UniRef90 (Suzek *et al.*, 2015) and Metaclust (Steinegger and Söding, 2018)] through a combination of the HH-suite and HMMER programs (Fig. 1).

In Stage 1 (Fig. 1 first column), HHblits from HH-suite 2.0.16 is used to search UniClust30 with the parameters '-diff inf -id 99 -cov 50 -n 3'. After testing HHblits MSA generated using the last version of UniProt20 (2016_02), latest Uniboost30 (2016_09) and three recent versions of Uniclust30 (2017_04, 2017_07, 2017_10), we found the three versions of Uniclust30 generate MSAs with comparable quality, all with a higher contact prediction accuracy than MSA

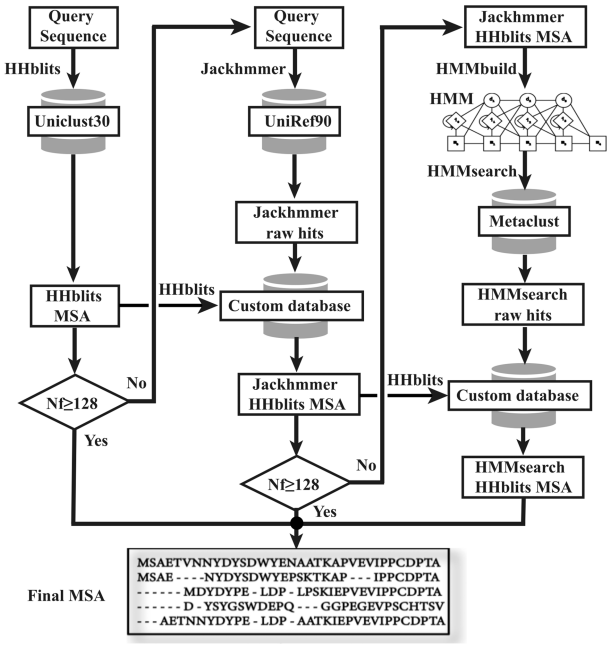

Fig. 1. Flowchart of DeepMSA. Three stages of MSA generations are performed consecutively using sequences from HHblits search through Uniclust30 (first column), Jackhmmer through UniRef (second column) and HMMsearch through Metaclust (third column)

generated by either UniProt20 or Uniboost30. Therefore, an arbitrary UniClust30 version (2017_10) is used for this study.

If Stage 1 does not generate enough sequences, i.e. Nf < 128, Stage 2 will be performed (Fig. 1 second column), where Jackhmmer is used to search against UniRef90 with parameters '-N 3 -E 10 –incE 1e-3'. We choose '-E 10' because lowering this *e*-value cutoff occasionally results in the inclusion of excessive number of non-homologous multi-domain hits in edge cases, although the final number of significant hits in the Jackhmmer alignment is determined by '–incE'. Instead of directly using the alignment generated by Jackhmmer search, esl-sfetch from the HMMER package is used to extract full-length sequences according to the list of Jackhmmer hits. These full-length sequences are converted into a custom HHblits format database by 'hhblitdb.pl' script from HH-suite. After the construction of the custom database, HHblits is again applied to search this custom database using the same search parameter as in Stage 1 but jump-starting the search from the Stage 1 sequence MSA. If the MSA from Stage 2 has an Nf higher than that from Stage 1 MSA, it will replace the Stage 1 MSA for subsequent computation.

DeepMSA implements two time-saving heuristics to reduce time complexity associated with construction of HHblits format database, which, unlike conventional sequence databases, comprise of sequence profiles. Each profile can be either one sequence or one MSA within a family of protein sequences clustered by sequence identity. The time required to construct a profile database is proportional to the number of profiles and the average number of positions of the profiles. It may take many hours to construct a custom HHblits database if the sequences are very long or if there are too many sequences. To shorten the time for database construction, we trim the Jackhmmer hits and perform sequence clustering. In particular, instead of using the full-length Jackhmmer hit, we trim the Jackhmmer hit to extract the local region aligned to the query in the Jackhmmer alignment, as well the L flanking residues at both sides of the aligned regions. Moreover, all trimmed hits from the previous step are further clustered by kClust (Hauser *et al.*, 2013) into sequence clusters by 30% sequence identity cutoff. Next, Clustal Omega (Sievers *et al.*, 2011) is then used to align sequences within each cluster into aligned sequence profiles. These profiles are fed into hhblitsdb.pl to construct the custom HHblits database. As kClust and Clustal Omega usually take only a few minutes, and the

Table 1. Nf and the number of aligned homologous sequences (*N*) in the MSAs collected by different schemes

Schemes[a]	'Hard' targets		'Easy' targets		All targets	
	Nf	*N*	Nf	*N*	Nf	*N*
DeepMSA	119.67	3046.16	435.52	8869.82	331.20	6868.53
Stage 1	82.22	1698.12	430.49	8765.65	310.81	6336.91
Stage 2	131.30	3158.46	612.83	14 816.79	447.35	10 810.43
Stage 3	346.02	8098.61	1031.95	24 194.26	796.23	18 663.02
Jackhmmer	174.64	3720.27	727.95	17 818.32	537.81	12 973.55
PSI-BLAST	145.02	5032.81	739.06	21 195.11	534.92	15 640.96
No custom db	516.27	11 751.12	1642.74	49 326.13	1255.63	36 413.55

[a]Stages 1, 2 and 3 are three stages of DeepMSA. 'No custom db' modifies DeepMSA pipeline by directly concatenating HMMER alignments without custom HHblits database construction in Stages 2 and 3. 'PSI-BLAST' and 'Jackhmmer' search UniRef90 with PSI-BLAST and Jackhmmer, respectively.

number sequences is ~10 times larger than the number of kClust sequence clusters, it will take less than half an hour to construct the custom database.

If the MSA from previous stages still has Nf $<$ 128, Stage 3 is performed (Fig. 1 third column), where the MSA from the previous stage is converted into a HMM by HMMbuild from the HMMER package. This HMM is searched against Metaclust metagenome sequence database by HMMsearch, using parameters '-E 10 –incE 1e-3'. Similar to Stage 2, sequence hits from HMMsearch are built into a custom HHblits database. The MSA from previous stages is used to jump-start an HHblits search against this new custom HHblits database to derive the final Stage 3 MSA.

3 Results

3.1 Dataset

DeepMSA is tested on a set of 614 non-redundant proteins curated from the SCOPe database (Hubbard *et al.*, 2010) according to the following criteria: (i) any target coming from a fold with only one superfamily is excluded, because such a target is unlikely to have any remote structure analog; (ii) redundant sequences with a 30% pair-wise sequence identity are removed; (iii) each query should have at least one template structure, detectable by TM-align (Zhang and Skolnick, 2005), from the PDB which has a TM-score $>$0.5 with the sequence identity $<$0.3 to the query. These resulted in 614 proteins, which are classified into 403 'Easy' and 211 'Hard' targets by the meta-threading program, LOMETS (Wu and Zhang, 2007), based on the significance of threading alignments between query and template sequences. While our discussions are mainly focused on the 'Hard' targets which DeepMSA aims to address, the results for the 'Easy' targets are listed in the Supplementary Material for the completeness of comparisons.

3.2 Coverage and depth of MSAs by DeepMSA

Since one of the initial motivations for DeepMSA to combine sequences from different sequence databases is to collect more diverse sequences, it is instrumental to examine the coverage and depth of the MSA brought by DeepMSA. To this end, Table 1 lists the depth results of MSAs generated by six different schemes, including DeepMSA, its three stages and three baseline methods. Here, to obtain data for different stages, we force DeepMSA to perform all three stages regardless of Nf cutoff. Nevertheless, the final MSA in DeepMSA is calculated as the normal procedure, i.e. having the MSA constructed from Stage 1 if its Nf $\geq$ 128; or from Stage 2 if Stage 1 has Neff $<$ 128 but Stage 2 has Nf $\geq$ 128; or from Stage 3, otherwise. Two of the baseline methods generate MSAs by Jackhmmer or PSI-BLAST search against the same UniRef90 database as used by DeepMSA. For the last baseline method, denoted as 'No custom db' in Table 1, the custom HHblits database construction and HHblits search in Stages 2 and 3 are replaced by direct concatenation of HMMER (Jackhmmer and HMMsearch) MSAs to the MSA from the previous stage, similar to the approach reported earlier (Ovchinnikov *et al.*, 2017).

As expected, the alignment depth, when measured by Nf and the total number of detected sequences, gradually increases from Stage 1 to Stage 3. The increase is particularly large for 'Hard' targets, where the final MSAs from DeepMSA are on average 1.5 and 1.8 times deeper than Stage 1 in terms of Nf and number of sequences, respectively. On the other hand, the alignment depth of DeepMSA is significantly smaller than 'No custom db' and 'Jackhmmer'. This is because all HMMER hits are included in the 'No custom db' and 'Jackhmmer' alignments, while many HMMER hits are discarded by DeepMSA during HHblits search through custom databases.

It should be noted that the full-length MSA constructions often cost more memory and slow down the computing processes. Moreover, due to the composite profile construction and alignment algorithms, MSAs with greater Nf and sequence numbers do not necessarily indicate better MSA quality, as shown in later sections. In fact, there is no single index which can directly assess the performance of MSA collection programs. To more objectively assess the quality of MSA builders, below we apply these MSAs to three protein structure modeling experiments, i.e. residue contact prediction, SS prediction and protein fold-recognition (i.e. threading).

3.3 DeepMSA increases contact prediction accuracy

The utility of DeepMSA for contact prediction is assessed using six state-of-the-art programs: CCMpred (Seemayer *et al.*, 2014), MetaPSICOV2 (Buchan and Jones, 2018), DeepContact (Liu *et al.*, 2018), DeepCov (Jones and Kandathil, 2018), PConsC4 (Michel *et al.*, 2018) and TripletRes (Li *et al.*, 2019). Here, CCMpred is a representative coevolution-only contact predictor. MetaPSICOV2 is based on traditional (shallow and fully-connected) neural networks. The rest of the programs are based on deep convolutional neural networks. While other predictors with good performance also exist, we selected the six programs partly because of the availability of standalone packages, which facilitate the large-scale implement and comparison of the results.

In Table 2, we list the results of contact predictions by the six predictors, each having the MSA collected from the six schemes listed in Table 1. Since MetaPSICOV2 and DeepContact have their own built-in MSA generation protocols, both of which combine HHblits and jackhammer, contact precisions from the built-in MSAs are listed as 'default' in Table 2.

Here, as in community-wide Critical Assessment of protein Structure Prediction (CASP) challenges (Schaarschmidt *et al.*, 2018), a contact is defined as Cβ atoms (Cα atoms for glycine) from a pair of residues, i and j, being close to each other by $<$8 Å. Contact prediction accuracies of different methods are evaluated by precisions of top L, $L/2$ and $L/5$ medium-range ($12 \leq i-j \leq 23$) and long-range ($24 \leq i-j$) predicted contacts. In accordance with CASP convention, Table 2 only lists the long-range contacts of 'Hard' targets, where. For completeness, the results for medium-range contacts for all targets ('Hard' and 'Easy') are listed in Supplementary

Table 2. Long-range contact prediction precision for 211 'Hard' protein targets

Contact predictor	MSA	Top *L*	*P*-value	Top *L*/2	*P*-value	Top *L*/5	*P*-value
CCMpred	DeepMSA	0.268	*	0.375	*	0.483	*
	Stage 1	0.215	3.73E-24	0.307	4.78E-23	0.410	4.21E-15
	Stage 2	0.237	2.49E-13	0.333	1.19E-14	0.430	3.45E-13
	Stage 3	**0.280**	1.00	**0.381**	0.98	**0.486**	0.79
	Jackhmmer	0.227	3.84E-15	0.317	2.37E-15	0.418	1.54E-11
	PSI-BLAST	0.208	3.35E-24	0.289	2.18E-26	0.394	5.81E-16
	No custom db	0.264	0.187	0.366	4.83E-2	0.468	1.86E-2
Meta-PSICOV2	DeepMSA	0.410	*	0.532	*	**0.654**	*
	Stage 1	0.373	6.66E-13	0.483	1.32E-12	0.595	1.19E-10
	Stage 2	0.388	1.43E-6	0.501	2.25E-7	0.618	6.56E-6
	Stage 3	**0.412**	0.93	**0.534**	0.74	0.653	0.67
	Default	0.387	4.75E-5	0.500	1.79E-5	0.612	2.11E-5
	Jackhmmer	0.377	2.27E-7	0.490	1.24E-6	0.604	1.07E-5
	PSI-BLAST	0.336	1.46E-19	0.441	6.32E-16	0.546	4.42E-13
	No custom db	0.400	3.29E-2	0.515	1.43E-2	0.629	7/03E-3
Deep-Contact	DeepMSA	0.485	*	0.630	*	**0.756**	*
	Stage 1	0.445	3.43E-15	0.581	4.00E-13	0.716	3.60E-7
	Stage 2	0.458	5.07E-10	0.598	3.15E-8	0.730	7.63E-5
	Stage 3	**0.488**	0.99	**0.632**	0.92	0.754	0.13
	Default	0.434	1.37E-13	0.562	1.75E-13	0.681	5.35E-10
	Jackhmmer	0.441	1.07E-11	0.576	7.55E-10	0.702	2.76E-6
	PSI-BLAST	0.427	1.99E-16	0.553	6.42E-15	0.681	2.77E-9
	No custom db	0.472	1.84E-3	0.614	1.88E-3	0.732	5.17E-3
DeepCov	DeepMSA	**0.439**	*	**0.588**	*	**0.738**	*
	Stage 1	0.408	6.01E-9	0.553	6.85E-7	0.701	3.36E-5
	Stage 2	0.420	1.03E-5	0.561	3.51E-6	0.712	5.46E-5
	Stage 3	0.439	0.49	0.586	0.35	0.730	9.68E-3
	Jackhmmer	0.392	1.21E-11	0.521	4.80E-11	0.662	2.28E-9
	PSI-BLAST	0.377	2.96E-18	0.505	7.01E-17	0.649	5.16E-12
	No custom db	0.421	7.09E-4	0.563	1.61E-3	0.708	2.21E-3
PConsC4	DeepMSA	0.475	*	0.610	*	0.718	*
	Stage 1	0.420	6.64E-17	0.544	4.10E-13	0.653	1.04E-7
	Stage 2	0.443	1.19E-8	0.572	5.52E-7	0.681	3.24E-4
	Stage 3	**0.478**	0.97	**0.612**	0.75	**0.719**	0.70
	Jackhmmer	0.420	2.08E-11	0.545	1.61E-8	0.652	3.69E-6
	PSI-BLAST	0.364	8.64E-16	0.474	2.89E-14	0.572	4.55E-12
	No custom db	0.462	1.09E-2	0.593	2.38E-2	0.697	3.72E-2
TripletRes	DeepMSA	**0.610**	*	**0.759**	*	**0.860**	*
	Stage 1	0.594	6.37E-6	0.742	5.78E-4	0.849	2.59E-2
	Stage 2	0.601	2.65E-4	0.747	6.65E-4	0.856	0.17
	Stage 3	**0.610**	0.34	0.756	8.34E-2	0.859	0.29
	Jackhmmer	0.565	3.11E-8	0.704	1.00E-7	0.815	9.40E-5
	PSI-BLAST	0.547	1.35E-13	0.684	2.15E-13	0.790	2.50E-9
	No custom db	0.584	1.83E-5	0.728	8.00E-5	0.830	7.85E-4

Note: Bold font indicates the highest value in each category. The standard deviation of the average precision is presented in Supplementary Table S4.

*Each *P*-value is calculated by one-tailed paired *t*-test to test whether DeepMSA has significant higher contact prediction accuracy than the respective MSA.

Table S1. We also provide a spreadsheet file for per-target assessment result in Supplementary Table S4.

It is shown that the MSA from DeepMSA outperforms the default MSA for contact prediction in all six contact predictors. For instance, the precisions for the top *L* contacts generated by TripletRes and CCMpred increased by 2.7 and 24.4%, respectively, when they use the MSA from DeepMSA instead of the default MSA. Furthermore, contact precision improves progressively from Stage 1 to Stage 3 for all the programs, indicating the effectiveness of depth of MSAs in contact prediction. Contact precisions from DeepMSA are also consistently higher than those from HHblits (i.e. Stage 1), Jackhmmer and PSI-BLAST alone.

We note that the output MSA of DeepMSA is not always created from Stage 3 if previous two stages achieve Nf $\geq$ 128, which helps to save the memory and running time of DeepMSA. Interestingly, this setting does not degrade contact precision significantly for most predictors. In fact, for TripletRes and DeepCov, the MSA from DeepMSA yields slightly better contact precision compared to the MSA from DeepMSA Stage 3. Figure 2 shows the effect of Nf cutoff in DeepMSA on the precision of contact prediction, where, for all but one program (CCMpred), increasing the Nf cutoff over 128 has no obvious improvement on contact precisions. In other words, when the alignment is already deep (Nf $\geq$ 128), further inclusion of more sequences is indeed not beneficial for all five neural network-based contact predictors. This might be because deeper MSAs are more prone to contain alignment errors and false positive hits, where the cutoff of Nf = 128 might be the result of the tradeoff between the sequence coverage and alignment noises. Moreover, this result may also suggest that the sequence datasets from the standard Uniclust30 utilized in Stage 1 is more reliable than the UniRef90

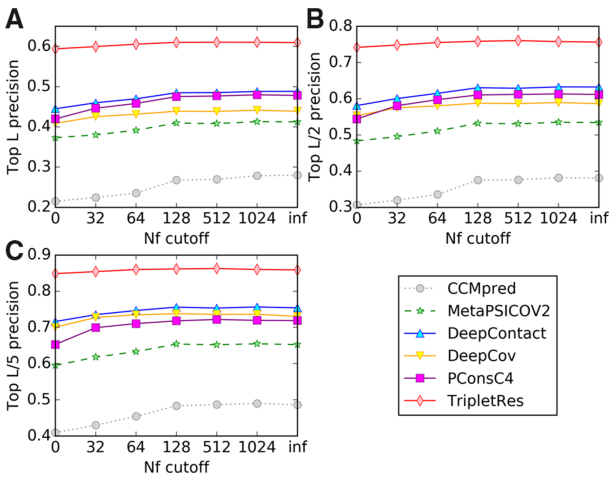

Fig. 2. Nf cutoff of DeepMSA versus top *L* (**A**), top *L*/2 (**B**) and top *L*/5 (**C**) long-range contact prediction precision. The Nf cutoff of 0 and inf correspond to always using Stage 1 and Stage 3 MSAs, respectively

and metagenomic database, and thus the addition of more sequences from the latter datasets might have the tendency to introduce more noises.

We note that the high quality of MSA from DeepMSA is not merely the result of combining multiple sequence databases. In particular, apart from the lack of custom HHblits database construction and search step, 'No custom db' uses identical sequence databases, with the same HHblits and HMMER programs as DeepMSA. Despite ~4 times greater alignment depth (Table 1), 'No custom db' is worse than DeepMSA by 1.0% (CCMpred) to 4.2% (TripletRes) in terms of top *L* contact precision (Table 2). These data suggest again that deeper alignments (with more sequence homologs) do not necessarily guarantee better contact prediction. It also indicates that although diverse sequence databases are contributive to DeepMSA performance, it is also essential to combine multiple sequence search and alignment algorithms, especially the custom HHblits database construction subroutines in our case.

DeepMSA also outperforms the default MSAs in DeepContact and MetaPSICOV. In particular, the Stage 2 MSA yields slightly more precise (0.3%) top *L* contact prediction by MetaPSICOV than its default MSA, even though both kinds of MSAs come from HHblits search through custom HHblits database constructed from Jackhmmer hits. This show that our time-saving heuristics (HMMER hit trimming and kClust clustering, which result in an overall average DeepMSA running time of 0.7 h per protein, Supplementary Fig. S2) introduce little compromise to final alignment quality.

Apart from benchmark data discussed herein, DeepMSA was also blindly tested in CASP13 as the MSA generation pipeline for our TripletRes server (Li *et al.*, 2019), whose average top *L* contact precisions on all 31 FM targets increased from 0.332 with HHblits MSAs to 0.409 with DeepMSA.

3.4 DeepMSA enables more accurate threading

Threading is an important approach to template-based protein structure prediction, which recognizes proteins with similar fold to the query proteins. Since most of the state-of-the-art methods use profiles, in the form of either HMM or position specific scoring matrix, to deduce query-template alignments, we examine whether and how DeepMSA can impact the performance of two typical threading programs, HHsearch (Soding, 2005) and MUSTER (Wu and Zhang, 2008), which by default use HHblits and PSI-BLAST to construct sequence profile, respectively.

The HHsearch and MUSTER template database is constructed from the 71 684 non-redundant (pair-wise sequence identity <70%) protein structures from the I-TASSER (Yang *et al.*, 2015) template library at https://zhanglab.ccmb.med.umich.edu/library/. To generate the HHsearch library with default profile and with our new profiles, we first build MSAs for all templates by HHblits search against Uniclust30 database and DeepMSA, respectively. The hhmake program from HH-suite is then used to convert the MSAs to HHsearch style HMM library.

Table 3. Benchmark results for the first threading template on 211 'Hard' targets

Method	TM-score	*P*-value	RMSD (Å)	Coverage	#(TM-score >0.5)
HHsearch	0.308	5.70E-03	11.15	0.665	33
HHsearch[a]	**0.331**	*	11.17	0.697	**46**
MUSTER	0.311	7.40E-04	13.62	0.872	25
MUSTER[a]	**0.345**	*	12.87	0.851	**41**

Note: Bold font indicates the highest value in each category.

[a]Indicates threading with DeepMSA profile.

*Each *P*-value is calculated by one-tailed paired *t*-test to test whether DeepMSA results in significantly more accurate threading result than the default profile.

In MUSTER, the default sequence profiles are constructed by searching NR database with blastpgp, i.e. the legacy PSI-BLAST program (Altschul *et al.*, 1997). Checkpoint files from PSI-BLAST search is then converted to MTX format sequence profiles. Conversion of DeepMSA alignments to MTX format is implemented by the 'a3m2mtx.pl' script in the DeepMSA package. This script jump-starts a PSI-BLAST search using the MSA of DeepMSA against a dummy BLAST format database. The MTX file can then be recovered from the checkpoint file of the jump-start search. Similarly, for query proteins, we also construct both DeepMSA profiles and default profiles.

In Table 3, we list a comparison of template alignments obtained by HHsearch and MUSTER using different MSAs. The results are presented only for 'Hard' targets in terms of the average TM-score (Zhang and Skolnick, 2004), alignment coverage (number of aligned residues divided by query length) and RMSD of aligned regions, where all templates with a sequence identity >30% to the query have been excluded. The results for 'Easy' and all targets are listed in Supplementary Table S2. It is shown that, for 'Hard' threading targets, the TM-score of first template by MUSTER and HHsearch is increased by 10.9 and 7.5%, respectively, if the DeepMSA profiles instead of the default PSI-BLAST/HHblits profiles are used. Of note, the number of 'Hard' targets with correctly identified templates (TM-score >0.5) is increased by 64.0 and 39.4% for MUSTER and HHsearch, respectively.

The observation that DeepMSA significantly boosts threading performance for 'Hard' targets can be partially explained by improved quality of query-template alignments. To examine this point, we curate a subset of 143 'enriched' 'Hard' targets, each of them having at least 30 templates of the correct fold (TM-score >0.5) detectable by TM-align with <30% sequence identity to the query. For each of these targets, we calculate average TM-score with all the templates aligned by HHsearch using DeepMSA sequence profile and compare it to that using the default HHblits profile used by HHsearch. Figure 3A lists the average TM-score difference on the top 30 templates for each of 143 targets. The data show that DeepMSA generated positive impact on the query-template alignments for 68.5% (=98/143) of the cases. Among the 98 cases, 69 (70.4%) have the TM-score difference with *P*-value <0.05 in the paired *t*-test (dark bars in Fig. 3A), showing that the difference is statistically significant although only about 30 data points are involved in the paired *t*-test calculation for each target.

To further illustrate the importance of DeepMSA profile in threading, we show a case study on query d1hx6a2 and its template 2bbdA. HHsearch threading based on DeepMSA profile correctly aligns query to C-terminal (residue 167 to 319) of template and achieves a TM-score =0.61 (Fig. 3B); the alignment region is similar to that by the structure alignment from TM-align, although TM-align

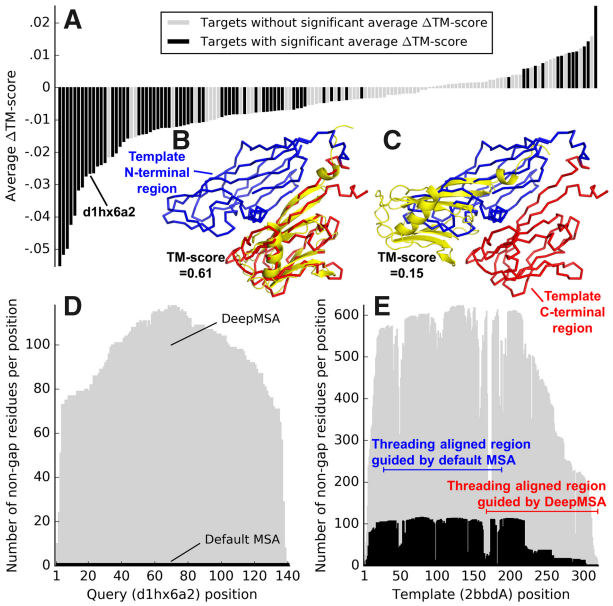

Fig. 3. Contribution of DeepMSA to query-template alignment in HHsearch threading. (**A**) For each template of a 'Hard' target, we calculate the TM-score of HHsearch guided by default profile minus that by DeepMSA profile (ΔTM-score). The *y*-axis is the average ΔTM-score for each target, ranked in ascending order (*x*-axis) of average ΔTM-score. To calculate the statistical significance of ΔTM-score for each target, we perform a paired *t*-test between TM-score pairs (i.e. TM-score by DeepMSA versus TM-score by default profile) for all templates of the target. Targets with significant ΔTM-score are colored in black. (**B, C**) Alignment of query d1hx6a2 (cartoon) to template 2bbdA (upper left and lower right ribbons for N and C-terminal regions, respectively) using DeepMSA profile (B) and default profile (C). (**D, E**) Number of non-gap residues (*y*-axis) at each position (*x*-axis) in the DeepMSA profile (grey) and in the default HHblits profile (black) for query (d1hx6a2) (D) and template (2bbdA) (E)

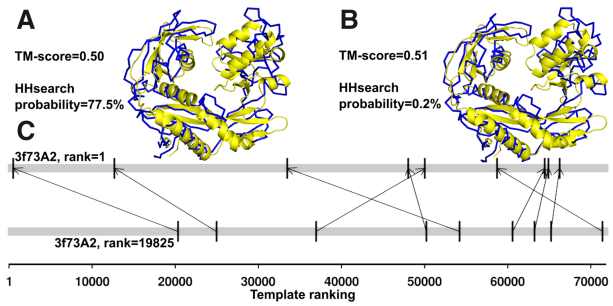

Fig. 4. Contribution of DeepMSA to HHsearch template ranking for query d1yvua1. (**A** and **B**) Threading alignment between query (cartoon) and template 3f73A2 (ribbon), guided by DeepMSA profile (A) and by default HHblits profile (B). (**C**) Ranking of nine correct templates (TM-score >0.5, black vertical lines) among all 70 977 templates (grey horizontal bands) after excluding template proteins with a sequence identity >30% to the query. Template rankings guided by DeepMSA profile and that by the default profile are shown in upper and lower bands, respectively. The same template in the two cases is connected by a thin arrow

has an even higher TM-score (=0.82, see Supplementary Fig. S3). On the other hand, HHsearch threading with the default HHblits profile only gets a TM-score =0.15 due to complete mis-alignment of query to the N-terminal (residue 27 to 188) of template (Fig. 3C). Such differences can be explained by depths of MSAs for both query and template: the default HHblits run only detects 133 homologs for the template and no homolog for the query. On the other hand, DeepMSA profile is much deeper, with 624 and 118 homologs for the template (Fig. 3D) and the query (Fig. 3E), respectively. The lack of template homologs in the default run is particularly severe at the C-terminal of the template, driving HHsearch to align the query to the template N-terminal instead.

Table 4. Summary of SS prediction by PSSpred for 211 'Hard' targets

MSA	Q3	*P*-value	SOV	*P*-value
PSI-BLAST + UniRef90	80.518	1.38E-03	77.257	1.05E-03
DeepMSA	**81.472**	*	**78.660**	*

Note: Bold font indicates the higher value in each category.

*Each *P*-value is calculated by one-tailed paired *t*-test to test whether DeepMSA results in significantly more accurate SS prediction than the default profile.

In addition to the creation of correct alignments, another reason for the performance improvement by DeepMSA on threading is that better MSA profiles can help improve the ranking of the template alignments. In Figure 4, we show an example from the query protein (d1yvua1) which is aligned on the template 3f73A2 using HHsearch. Although both default and DeepMSA profiles resulted in reasonable query-template alignments with a TM-score >0.5, their alignment scores are very different. While the HMM probability on the DeepMSA profile is 77.5% which puts the template as ranked No. 1, the probability score is 0.2% using the default profile which is ranked at 19 825th position among all templates. Thus, although the default profile can generate correct alignment on this query-template pair, the correct template cannot be selected by the threading program due to the poor alignment scores. In this case, an unrelated protein (3iz6D3, TM-score =0.08) was selected as the first template when using the default HMM profile alignments.

3.5 DeepMSA profiles improve SS prediction over traditional PSI-BLAST profiles

In this section, we further test the performance of DeepMSA in SS prediction by PSIPRED 4.0 (Jones, 1999) and PSSpred (Yan *et al.*, 2013). By default, PSIPRED and PSSpred construct MTX format sequence profiles by searching UniRef90 or NR database with PSI-BLAST program (Altschul *et al.*, 1997). MTX format DeepMSA profile for these two programs can also be obtained by a3m2mtx.pl.

The accuracy of the SS predictions by PSSpred (Table 4) and PSIPRED (Supplementary Table S3) is evaluated by Q3 accuracy and SOV segment overlap measure (Zemla *et al.*, 1999). Compared to the default profiles, sequence profiles from DeepMSA improve the Q3 accuracy by 1.2 and 1.0% for PSSpred and PSIPRED, respectively. Similarly, SOV scores by PSSpred and PSIPRED are improved by 1.8 and 1.5%, respectively, when MSAs from DeepMSA are used. The differences are statistically significant, since the *P*-values in Student's *t*-test are all below 0.002.

Here, it important to note that the original models of PSSpred and PSIPRED were trained based on 2011 and 2016 sequence databases, respectively. Although SS predictions, as well as the contact and threading programs studied in previous sections, are usually sensitive to the sequence databases and MSAs that the models are originally trained on, we do not attempt to re-train the models using the new DeepMSA profiles. In this context, the performance improvement should be mainly attributed to the sensitive and comprehensive information that DeepMSA provides, compared to the MSAs generated by other default programs.

4 Conclusion

We developed an open-source pipeline, DeepMSA, aiming to collect deep and sensitive MSAs from whole-genome and metagenome sequence databases. Large-scale benchmark experiments show that DeepMSA consistently improves protein contact prediction, fold-recognition and SS prediction, compared to the widely used HHblits, Jackhmmer and PSI-BLAST sequence searching programs. For example, the use of MSAs from DeepMSA improves

top *L* long-range contact prediction precision of CCMpred by 24.4% compared to the default use of the HHblits MSAs by the program. Similarly, MUSTER threading identifies correct templates for 64.0% more 'Hard' targets by switching the default PSI-BLAST profiles to the DeepMSA profiles. Notably, all improvements in contact prediction, SS prediction and threading have been achieved without re-training predictor model and parameters in neural networks or dynamic programing alignment.

The high quality of MSA by DeepMSA is partly due to the greater coverage and alignment depth resulted from the combination of diverse source of sequence databases. However, benchmark study shows that deeper MSA with more sequence homologs does not always lead to better contact prediction, since the final effect of MSAs is often a tradeoff of sequence coverage and alignment accuracy. Further analysis reveals that appropriate incorporation of multiple sequence search and alignment algorithms is the key to generate high quality MSAs by DeepMSA. In particular, HMMER alignment reconstruction by custom HHblits database generation is found to be especially helpful: a baseline method ('No custom db' in Tables 1 and 2) without the custom HHblits database generation step results in 1.0–4.2% worse top *L* long-range contact prediction accuracies than DeepMSA, even when both methods use identical sequence databases.

The on-line server and the standalone program of DeepMSA have been made freely available at https://zhanglab.ccmb.med.umich.edu/DeepMSA/. The continuous developments of robust MSA and profile construction methods should help enhance the usefulness and impacts of the whole-genome and metagenomics initiatives on the structure and function prediction studies of the community. For example, the current DeepMSA program runs only with monomer proteins, while an extension of the program for protein–protein complex MSA constructing is important and under progress.

Acknowledgements

We thank Dr Xiaoqiong Wei and Qiqige Wuyun for insightful discussions, Robin Pearce for the original design of Figure 1 and Dr Sean R. Eddy for technical help regarding HMMER programs. This work used the Extreme Science and Engineering Discovery Environment (XSEDE) (Towns *et al.*, 2014), which is supported by National Science Foundation (ACI-1548562).

Funding

This work was supported by the National Institutes of Health [GM083107, GM116960, AI134678 to Y.Z.]; and the National Science Foundation [DBI1564756, IIS1901191 to Y.Z.].

Conflict of Interest: none declared.

References

Adhikari,B. *et al.* (2018) DNCON2: improved protein contact prediction using two-level deep convolutional neural networks. *Bioinformatics*, **34**, 1466–1472.

Altschul,S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Buchan,D.W.A. and Jones,D.T. (2018) Improved protein contact predictions with the MetaPSICOV2 server in CASP12. *Proteins*, **86**, 78–83.

Cozzetto,D. (2016) FFPred 3: feature-based function prediction for all Gene Ontology domains. *Sci. Rep.*, **6**, 31865.

Eddy,S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.

Gil,N. and Fiser,A. (2019) The choice of sequence homologs included in multiple sequence alignments has a dramatic impact on evolutionary conservation analysis. *Bioinformatics*, **35**, 12–19.

Hanson,J. *et al.* (2018) Accurate prediction of protein contact maps by coupling residual two-dimensional bidirectional long short-term memory with convolutional neural networks. *Bioinformatics*, **34**, 4039–4045.

Hauser,M. *et al.* (2013) kClust: fast and sensitive clustering of large protein sequence databases. *BMC Bioinformatics*, **14**, 248.

He,B. *et al.* (2017) NeBcon: protein contact map prediction using neural network training coupled with naïve Bayes classifiers. *Bioinformatics*, **33**, 2296–2306. btx

Hubbard,T.J.P. *et al.* (2010) SCOP, Structural Classification of Proteins Database: applications to Evaluation of the Effectiveness of Sequence Alignment Methods and Statistics of Protein Structural Data. *Acta Cryst.*, **54**, 1147–1154.

Jones,D.T. (1999) Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.*, **292**, 195–202.

Jones,D.T. and Kandathil,S.M. (2018) High precision in protein contact prediction using fully convolutional neural networks and minimal sequence features. *Bioinformatics*, **34**, 3308–3315.

Li,Y. *et al.* (2019) Ensembling multiple raw coevolutionary features with deep residual neural networks for contact-map prediction in CASP13. *Proteins*, **87**, 1082.

Liu,Y. *et al.* (2018) Enhancing evolutionary couplings with deep convolutional neural networks. *Cell Syst.*, **6**, 65–74.

Michel,M. *et al.* (2018) PconsC4: fast, free, easy, and accurate contact predictions. *Bioinformatics*, **35**, 2677–2679.

Mirdita,M. *et al.* (2017) Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic Acids Res.*, **45**, D170–D176.

Ovchinnikov,S. *et al.* (2018) Protein structure prediction using Rosetta in CASP12. *Proteins*, **86** (**Suppl. 1**), 113–121.

Ovchinnikov,S. *et al.* (2017) Protein structure determination using metagenome sequence data. *Science*, **355**, 294–298.

Remmert,M. *et al.* (2012) HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat. Methods*, **9**, 173–175.

Schaarschmidt,J. *et al.* (2018) Assessment of contact predictions in CASP12: co-evolution and deep learning coming of age. *Proteins*, **86** (**Suppl. 1**), 51–66.

Seemayer,S. *et al.* (2014) CCMpred—fast and precise prediction of protein residue–residue contacts from correlated mutations. *Bioinformatics*, **30**, 3128–3130.

Sievers,F. *et al.* (2011) Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol. Syst. Biol.*, **7**, 539.

Soding,J. (2005) Protein homology detection by HMM-HMM comparison. *Bioinformatics*, **21**, 951–960.

Steinegger,M. *et al.* (2019) HH-suite3 for fast remote homology detection and deep protein annotation. *BMC Bioinformatics*, **20**, 473.

Steinegger,M. and Söding,J. (2018) Clustering huge protein sequence sets in linear time. *Nat. Commun.*, **9**, 2542.

Suzek,B.E. *et al.* (2015) UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, **31**, 926–932.

Towns,J. *et al.* (2014) XSEDE: accelerating scientific discovery. *Comput. Sci. Eng.*, **16**, 62–74.

Wang,S. *et al.* (2017) Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS Comput. Biol.*, **13**, e1005324.

Wang,Y. *et al.* (2019) Fueling ab initio folding with marine microbiome enables structure and function predictions of new protein families. *Genome Biol.*, **20**, 229.

Wu,S. *et al.* (2011) Improving protein structure prediction using multiple sequence-based contact predictions. *Structure*, **19**, 1182–1191.

Wu,S. and Zhang,Y. (2007) LOMETS: a local meta-threading-server for protein structure prediction. *Nucleic Acids Res.*, **35**, 3375–3382.

Wu,S. and Zhang,Y. (2008) ANGLOR: a composite machine-learning algorithm for protein backbone torsion angle prediction. *PLoS One*, **3**, e3400.

Wu,S. and Zhang,Y. (2008) MUSTER: improving protein sequence profile–profile alignments by using multiple sources of structure information. *Proteins*, **72**, 547–556.

Yan,R. *et al.* (2013) A comparative assessment and analysis of 20 representative sequence alignment methods for protein structure prediction. *Sci. Rep.*, **3**, 2619.

Yang,J. *et al.* (2013) Protein-ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment. *Bioinformatics*, **29**, 2588–2595.

Yang,J. *et al.* (2015) The I-TASSER Suite: protein structure and function prediction. *Nat. Methods*, **12**, 7.

Zemla,A. *et al.* (1999) A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment. *Proteins*, **34**, 220–223.

Zhang,C. *et al*. (2017) COFACTOR: improved protein function prediction by combining structure, sequence and protein–protein interaction information. *Nucleic Acids Res*., **45**, W291–W299.

Zhang,C. *et al*. (2018) Template-based and free modeling of I-TASSER and QUARK pipelines using predicted contact maps in CASP12. *Proteins*, **86** (**Suppl. 1**), 136–151.

Zhang,Y. and Skolnick,J. (2004) Scoring function for automated assessment of protein structure template quality. *Proteins*, **57**, 702–710.

Zhang,Y. and Skolnick,J. (2005) TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res*., **33**, 2302–2309.

Zheng,W. *et al*. (2019) LOMETS2: improved meta-threading server for fold-recognition and structure-based function annotation for distant-homology proteins. *Nucleic Acids Res*., **47**, W429–W436.